CS771 Introduction to Machine Learning

Companion Arbiter PUF

1. Part 1

Let t_i^U denote the total time after which upper leaves the i^{th} multiplexer and t_i^L denote the time after which lower signal leaves the i^{th} multiplexer.(i = 0,1,2,...31).
Also, let a_i and b_i be the respective time taken by upper and lower signals while passing i^{th} multiplexer when the selected challenge bit(c_i) is 0. Let r_i and s_i be the respective time taken by upper and lower signals while passing i^{th} multiplexer when the selected challenge bit(c_i) is 1.
Note that,

$$t_i^U = (1 - c_i)(t_{i-1}^U + a_i) + c_i(t_{i-1}^L + s_i)$$
$$t_i^L = (1 - c_i)(t_{i-1}^L + b_i) + c_i(t_{i-1}^U + r_i)$$

Lets use the shorthand $\Delta_i = t_i^U - t_i^L$ to denote the time lag between upper and lower signal. Observe that,

$$\Delta_i = d_i\Delta_{i-1} + \alpha_i d_i + \beta_i$$

where, $d_i = 1 - 2c_i, \quad \alpha_i = (a_i - b_i + r_i - s_i)/2$ and $\beta_i = (a_i - b_i - r_i + s_i)/2$
Setting $\Delta_{-1} = 0$(that is absorbing the initail delays in a_0, b_0, r_0 *and* s_0), we obtain

$$\Delta_{31} = u_0x_0 + u_1x_1 + ... + u_{31}x_{31} + \beta_{31} = u^Tx + p \quad (1)$$

where, $u = (u_0, u_1, ..., u_{31})^T$

$$x_i = d_i.d_{i+1}....d_{31}$$
$$u_0 = \alpha_0$$
$$u_i = (\alpha_i + \beta_{i-1}) \quad (for \quad i > 0)$$

If Δ_{31}<0, upper signal wins and answer is 0.
If Δ_{31}>0, lower signal wins and answer is 1.
Thus, answer is simply -

$$(sign(u^Tx + p) + 1)/2.$$

Thus, the simple arbiter PUF with 32 bit challenge can be cracked by learning the linear model $\Delta_w = u^Tx+p = 0$
In similar manner, the reference model can be cracked by learning the linear model

$$\Delta_r = v^Tx + q = 0 \quad (2)$$

Let y denotes the response of challenge from the CAR-PUF, then

$$y = \begin{cases} 0; & |\Delta_w - \Delta_r| \leq \tau \\ 1; & |\Delta_w - \Delta_r| > \tau \end{cases}$$
$$= \begin{cases} 0; & |u^T x + p - v^T x - q| \leq \tau \\ 1; & \text{otherwise} \end{cases}$$
$$= \begin{cases} 0; & |(u-v)^T x + (p-q)| \leq \tau \\ 1; & \text{otherwise} \end{cases}$$
$$= \begin{cases} 0; & \left((u-v)^T x + (p-q)\right)^2 - \tau^2 \leq 0 \\ 1; & \text{otherwise} \end{cases}$$
$$= \begin{cases} 0; & \left((u-v)^T x\right)^2 + (p-q)^2 + 2(p-q)(u-v)^T x - \tau^2 \leq 0 \\ 1; & \text{otherwise} \end{cases}$$

Note that,

$$\begin{aligned}
\left((u-v)^T x\right)^2 &= \left((u-v)^T x\right)\left((u-v)^T x\right) \\
&= ((u_0 - v_0)x_0 + (u_1 - v_1)x_1 + ... + (u_{31} - v_{31})x_{31})\,((u_0 - v_0)x_0 + (u_1 - v_1)x_1 + ... + (u_{31} - v_{31})x_{31}) \\
&= (u_0 - v_0)^2 x_0^2 + (u_1 - v_1)^2 x_1^1 + ... + (u_{31} - v_{31})^2 x_{31}^2 + \\
&\quad 2(u_0 - v_0)(u_1 - v_1)x_0 x_1 + ... + 2(u_0 - v_0)(u_{31} - v_{31})x_0 x_{31} + \\
&\qquad 2(u_1 - v_1)(u_2 - v_2)x_1 x_2 + ... + 2(u_1 - v_1)(u_{31} - v_{31})x_1 x_{31} \\
&\qquad\quad + ... + 2(u_{30} - v_{30})(u_{31} - v_{31})x_{30} x_{31} \\
&= \sum_{i=0}^{31} (u_i - v_i)^2 x_i^2 + \sum_{\substack{i,j=0 \\ i<j}}^{31} 2(u_i - v_i)(u_j - v_j)x_i x_j \\
&= \sum_{i=0}^{31} (u_i - v_i)^2 + \sum_{\substack{i,j=0 \\ i<j}}^{31} 2(u_i - v_i)(u_j - v_j)x_i x_j; \qquad \textit{since, } x_i^2 = 1
\end{aligned}$$

Also,

$$2(p-q)(u-v)^T x = \sum_{i=0}^{31} 2(p-q)(u_i - v_i)x_i$$

Hence we get,

$$\begin{aligned}
\left((u-v)^T x\right)^2 + (p-q)^2 + 2(p-q)(u-v)^T x - \tau^2 &= \sum_{\substack{i,j=0 \\ i<j}}^{31} 2(u_i - v_i)(u_j - v_j)x_i x_j + \sum_{i=0}^{31} 2(p-q)(u_i - v_i)x_i \\
&\quad + \sum_{i=0}^{31} (u_i - v_i)^2 + (p-q)^2 - \tau^2 \\
&= w^t \phi(c) + b
\end{aligned}$$

where,
w is a 528×1 vector, with its k^{th} elements given by

$$w_k = \begin{cases} 2(u_0 - v_0)(u_{k-(0-1)} - v_{k-(0-1)}); k = 0, 1, 2, ..., 30 \\ 2(u_1 - v_1)(u_{k-(31-2)} - v_{k-(31-2)}); k = 31, 32, ..., 60 \\ 2(u_2 - v_2)(u_{k-(61-3)} - v_{k-(61-3)}); k = 61, ..., 89 \\ 2(u_3 - v_3)(u_{k-(90-4)} - v_{k-(90-4)}); k = 90, 91, ..., 117 \\ . \\ . \\ . \\ 2(u_{30} - v_{30})(u_{31} - v_{31}); k = 495 \\ 2(p-q)(u_{k-496} - v_{k-496}); k = 496, 498, ..., 527 \end{cases} \tag{3}$$

$\phi(c)$ is a map from $\{0,1\}^{32}$ to R^{528} with its k^{th} component given by

$$\phi_k(c) = \begin{cases} x_0 x_{k-(0-1)}; k = 0,1,2,...,30 \\ x_1 x_{k-(31-2)}; k = 31,32,...,60 \\ x_2 x_{k-(61-3)}; k = 61,...,89 \\ x_3 x_{k-(90-4)}; k = 90,91,...,117 \\ . \\ . \\ . \\ x_{30} x_{31}; k = 495 \\ x_{k-496}; k = 496,498,...,527 \end{cases} \tag{4}$$

and b is the bias term given by, $b = \sum_{i=0}^{31} (u_i - v_i)^2 + (p-q)^2 - \tau^2$

Thus,

$$y = \begin{cases} 0; & w^t \phi(c) + b \leq 0 \\ 1; & w^t \phi(c) + b > 0 \end{cases}$$

Or

$$y = \frac{1 + sign(w^t \phi(c) + b)}{2}$$

Therefore this CAR-PUF can be cracked by learning the linear model

$$w^t \phi(c) + b = 0$$

. H.P.

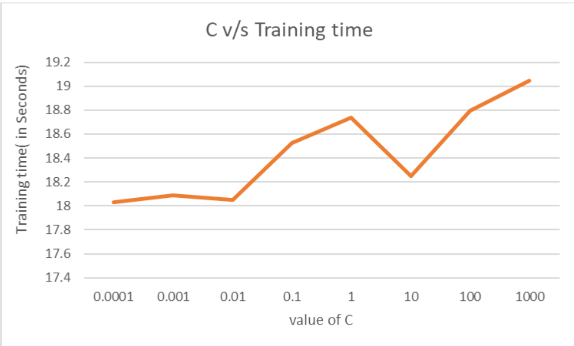
2. Part 3

For Logistic Regression

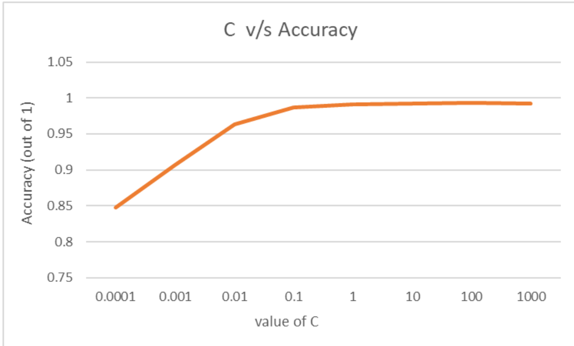
(i) Effect on the variation of C value

Value of C	Training time (s)	Test Accuracy (out of 1)
0.0001	18.03	0.8482
0.001	18.09	0.9069
0.01	18.05	0.9635
0.1	18.53	0.9871
1	18.74	0.9907
10	18.25	0.9922
100	18.80	0.9931
1000	19.05	0.9923

Table 1: Table for hyper Parameter C



(a) C vs. Training Time



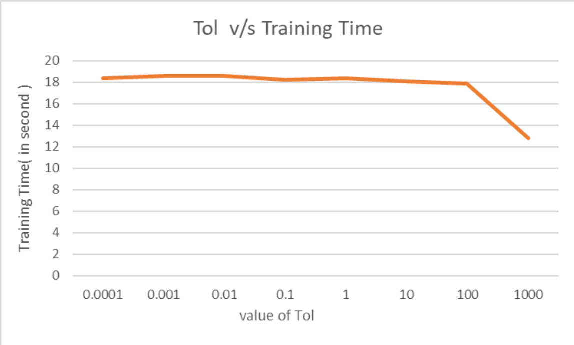
(b) C vs. Test Accuracy

Figure 1: Plots for Hyperparameter C

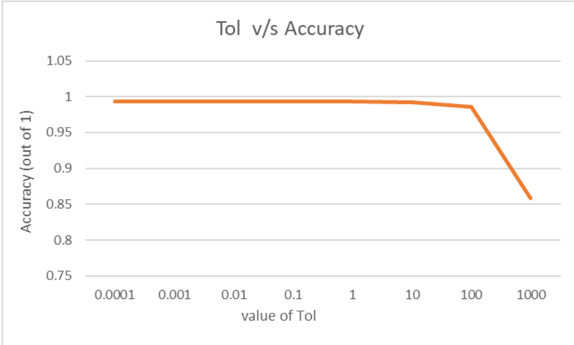
(ii) Effect on the variation of Tol value and fix the C=100 at which we found the max accuracy.

Value of Tol	Training time (s)	Test Accuracy(out of 1)
0.0001	18.34	0.9931
0.001	18.55	0.9931
0.01	18.56	0.9931
0.1	18.23	0.9931
1	18.38	0.9931
10	18.08	0.9923
100	17.85	0.9857
1000	12.80	0.8587

Table 2: Table for hyper Parameter Tol



(a) Tol vs. Training Time



(b) Tol vs. Test Accuracy

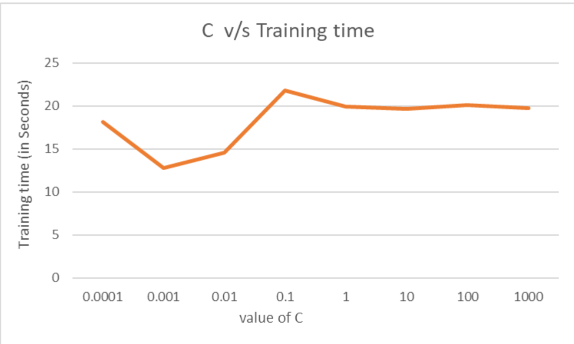
Figure 2: Plots for Hyperparameter Tol

for LinearSVC

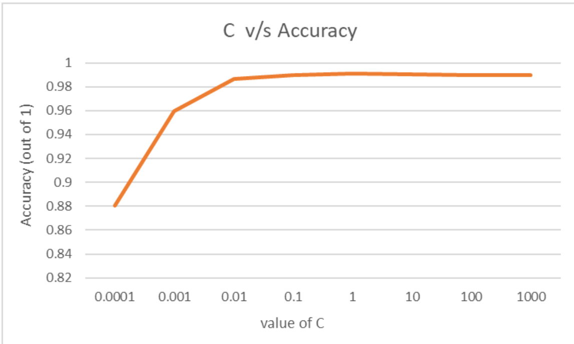
(i) Effect on the variation of C value

Value of C	**Training time (s)**	**Test Accuracy(out of 1)**
0.0001	18.09	0.8805
0.001	12.80	0.9597
0.01	14.59	0.9865
0.1	21.83	0.9899
1	19.89	0.99132
10	19.61	0.99014
100	20.13	0.98972
1000	19.77	0.98984

Table 3: Table for hyper Parameter C



(a) C vs. Training Time



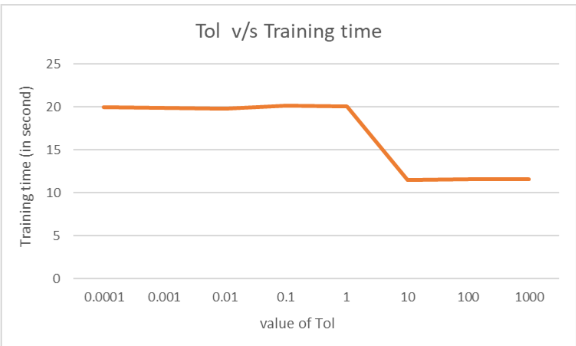
(b) C vs. Test Accuracy

Figure 3: Plots for Hyperparameter C

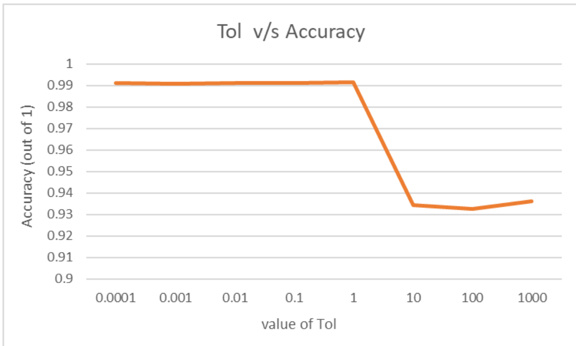
(ii) Effect on the variation of Tol value and fix the C=100 at which we found the max accuracy

Value of Tol	Training time (s)	Test Accuracy(out of 1)
0.0001	19.95	0.99124
0.001	19.86	0.9909
0.01	19.79	0.9912
0.1	20.17	0.99102
1	20.09	0.99136
10	11.48	0.93448
100	11.55	0.93252
1000	11.56	0.93604

Table 4: Table for hyper Parameter Tol



(a) Tol vs. Training Time



(b) Tol vs. Test Accuracy

Figure 4: Plots for Hyperparameter Tol